IN THE CLAIMS:

Please amend claims 1-4, 12 and 18, cancel claims 14-17 and 19, and add new claims 20-25 as follows.

1. (Currently amended) A method for securing a computer connected to an insecure network when the computer is not utilizing the insecure network, wherein the computer is installed with a program managing the connection with the insecure network, the method comprising the steps of:

determining whether the computer is active;

deactivating the computer from the insecure network when it is determined that the computer is inactive, thereby securing the computer; ~~and,~~

activating the computer to the network when it is determined that the computer is active; and

waiting for a predefined time period to repeat ~~the~~ said method from said step of determining whether the computer is active.

2. (Currently Amended) The method according to claim 1 further comprising the step of displaying the current status of the activation status of the insecure network on the computer.

3. (Currently amended) The method according to claim 1 further comprising the steps of:

obtaining an address for ~~the~~a network card;

obtaining an address for an interface connected to the insecure network using the obtained address of the network card; and,

obtaining the status of the obtained address of the interface.

4. (Currently Amended) The method according to claim 3 wherein said step of obtaining an address for the interface further comprises the steps of:

initializing any sockets support in the program managing the insecure network connection;

loading a driver having an object identifier of the program managing the insecure network connection;

obtaining an address for the initialization function and an address for the query function from the program; and,

calling the initialization function to initialize the driver.

5. (Original) The method according to claim 4 wherein said step of obtaining an address for an interface connected to the insecure network further comprises the steps of:

determining a total number of interface(s) using the obtained address of the network card; and,

storing the obtained total number of interface(s) in temporary memory.

6. (Original) The method according to claim 5 wherein said step of obtaining the status of each obtained address of the interface further comprises the steps of:

reading the status of the obtained address of the interface; and,

saving the obtained address of the interface with the read status to memory.

7. (Original) The method according to claim 3 wherein said step of deactivating the computer from the insecure network further comprises the step of setting each obtained address of the interface to an inactive status.

8. (Original) The method according to claim 1 further comprising the steps of:

determining whether there is a network reactivation request; and,

reactivating the computer on the insecure network when there is a network reactivation request.

9. (Original) The method according to claim 1 further comprising the steps of:

determining whether there is a network deactivation request; and,

deactivating the computer from the insecure network when there is a network deactivation request.

10. (Original) The method according to claim 3 wherein prior said step of determining whether the computer is active further comprises the steps of:

determining whether the obtained address of the interface connected to the insecure network has an active status; and,

waiting for a predefined time period to repeat the method when the obtained address of the interface has a nonactive status.

11. (Original) The method according to claim 1 wherein said step of determining whether the computer is active further comprises the steps of:

determining whether there is any active network process currently running via the insecure network when it is determined that the computer is active;

deactivating the computer from the insecure network when it is determined that there is no active network process currently running via the insecure network; and,

waiting for a predefined time period to repeat the method when it is determined that there is an active network process currently running via the insecure network.

12. (Currently amended) The method according to claim 10 wherein said step of determining whether there is any active network process currently running further comprises the steps of:

obtaining an address for ~~the~~a network card;

obtaining an address for an interface connected to the insecure network using the obtained address of the network card;

reading an old number of received and transmitted bytes over the obtained address of the interface;

changing the obtained address of the interface to an address for obtaining the number of bytes received;

reading the number of bytes received;

saving the read number of bytes received as a new number;

changing the obtained address of the interface to an address for obtaining the number of bytes transmitted;

reading the number of bytes transmitted;

saving the read number of bytes transmitted as a new number;

determining whether the old numbers of received and transmitted bytes equal to the new numbers of received and transmitted bytes;

returning a determination that an active network process is currently active when the old numbers do not equal the new numbers; and,

returning a determination that no active network process is currently running when the old numbers equal the new numbers.

13. (Original) The method according to claim 1 wherein said step of determining whether the computer is active is performed by a step of determining whether the screen saver is activated on the computer.

14. (Cancelled)

15. (Cancelled)

16. (Cancelled)

17. (Cancelled)

18. (Currently amended) A system for securing a computer connected to an insecure network when the computer is not utilizing the insecure network, wherein the computer is installed with a program managing the connection with the insecure network, the system comprising:

means for determining whether the computer is active;

means for deactivating the computer from the insecure network when it is determined that the computer is inactive, thereby securing the computer; ~~and,~~

means for activating the computer to the network when it is determined that the computer is active; and

means for waiting for a predefined time period to repeat the above method from the step of determining whether said computer is active.

19. (Cancelled)

20. (New) A method for securing a computer having at least one network interface connected to an insecure network when the computer is not utilizing the insecure network, the method comprising the steps of:

building an array of at least one network interface including a unique identifier for uniquely identifying each said at least one network interface and a status associated to each unique identifier for indicating the status of said unique identifier;

determining whether the computer is active or utilizing the insecure network;

turning off or deactivating the connection of the computer to the insecure network when it is determined that the computer is inactive, thereby securing the computer;

turning on or activating the connection of the computer to the inecure network when it is determined that the computer is active; and

waiting for a predefined time period to repeat from said step of determining whether the computer is active.

21. (New) The method according to claim 20 wherein prior to said step of building an array the method further comprises the steps of:

initializing any socket support managing the insecure network;

loading a driver having an object identifier managing the insecure network;

initializing commands of an Internet standard protocol; and

reading a configuration file for storing configuration information relating to the method.

22. (New) The method according to claim 20 wherein said step of building an array further comprises the steps of:

obtaining a total number of network interfaces available on the computer;

building an array of network interface indexes with a unique identifier for each said at least one network interface;

building an array of network interface types for each said unique identifier; and

building an array of network interface statuses for each said unique identifier.

23. (New) The method according to claim 22 wherein said step of building an array of network interface indexes further comprises the steps of:

obtaining a unique identifier for one of said at least one network interface;

storing the obtained unique identifier in the array;

determining whether additional ones of said at least one network interface are available;

if there are more said at least one network interface available, repeating from said step of obtaining a unique identifier for one of said at least one network interface; and

if there are no more said at least one network interface available, returning the array with the obtained unique identifier.

24. (New) The method according to claim 22 wherein said step of building an array of network interface types further comprises the steps of:

obtaining a network interface type for one of said unique identifier;

storing the obtained network interface type for said unique identifier in the array;

determining whether there are more said at least one network interface available;

if there are more said at least one network interface available, repeating from said step of obtaining a network interface type for one of said unique identifier; and

if there are no more said at least one network interface available, returning the array with the obtained network interface type.

25. (New) The method according to claim 22 wherein said step of building an array of network interface statuses to secure a computer further comprises the steps of:

obtaining a network interface status for one of said unique identifier;

storing the obtained network interface status for said unique identifier in the array;

determining whether there are more said at least one network interface available;

if there are more said at least one network interface available, repeating from said step of obtaining a network interface status for one of said unique identifier; and

if there are no more said at least one network interface available, returning the array with the obtained network interface status.